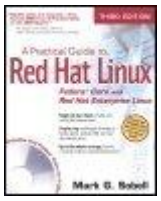


A Practical Guide to Red Hat® Linux®, Third Edition: Fedora™ Core and Red Hat Enterprise Linux

By Mark G. Sobell

.....

Publisher: **Prentice Hall**

Pub Date: **June 27, 2006**

Print ISBN-10: **0-13-228027-2**

Print ISBN-13: **978-0-13-228027-3**

Pages: **1168**

Table of Contents | Index

Overview

"Since I'm in an educational environment, I found the content of Sobell's book to be right on target and very helpful for anyone managing Linux in the enterprise. His style of writing is very clear. He builds up to the chapter exercises, which I find to be relevant to real-world scenarios a user or admin would encounter. An IT/IS student would find this book a valuable complement to their education. The vast amount of information is extremely well balanced and Sobell manages to present the content without complicated asides and meandering prose. This is a 'must have' for anyone managing Linux systems in a networked environment or anyone running a Linux server. I would also highly recommend it to an experienced computer user who is moving to the Linux platform." *Mary Norbury, IT Director, Barbara Davis Center/University of Colorado at Denver, from a review posted on slashdot.org*

"I had the chance to use your UNIX books when I when was in college years ago at Cal Poly San Luis Obispo, CA. I have to say that your books are among the best! They're quality books that teach the theoretical aspects and applications of the operating system." *Benton Chan, IS Engineer*

"The book has more than lived up to my expectations from the many reviews I read, even though it targets FC2. I have found something very rare with your book: It doesn't read like the standard a technical text, it reads more like a story. It's a pleasure to read and hard to put down. Did I say that?! :-)" *David Hopkins, Business Process Architect*

"Thanks for your work and for the book you wrote. There are really few books that can help people to become more efficient administrators of different workstations. We hope (in Russia) that you will continue bringing us a new level of understanding of Linux/UNIX systems." *Anton Petukhov*

"Mark Sobell has written a book as approachable as it is authoritative." *Jeffrey Bianchine, Advocate, Author, Journalist*

"Excellent reference book, well suited for the sysadmin of a Linux cluster, or the owner of a PC contemplating installing a recent stable Linux. Don't be put off by the daunting heft of the book. Sobell has striven to be as inclusive as possible, in trying to anticipate your system administration needs." *Wes Boudville, Inventor*

"*A Practical Guide to Red Hat® Linux®* is a brilliant book. Thank you Mark Sobell." *C. Pozrikidis, University of California at San Diego*

"This book presents the best overview of the Linux operating system that I have found. . . . It should be very helpful and understandable no matter what the reader's background is: traditional UNIX user, new Linux devotee, or even Windows user. Each topic is presented in a clear, complete fashion and very few assumptions are made about what the reader knows. . . . The book is extremely useful as a reference, as it contains a 70-page glossary of terms and is very well indexed. It is organized in such a way that the reader can focus on simple tasks without having to wade through more advanced topics until they are ready." *Cam Marshall, Marshall Information Service LLC, Member of Front Range UNIX Users Group FRUUG, Boulder, Colorado*

"Conclusively, this is THE book to get if you are a new Linux user and you just got into RH/Fedora world. There's no other book that discusses so many different topics and in such depth." *Eugenia Loli-Queru, Editor in Chief, OSNews.com*

The Best Just Became BETTERAgain!Completely Revised to Meet All Your Fedora Core and Red Hat Enterprise Linux Needs!

Fedora Core and Red Hat Enterprise Linux are advanced operating systems. You need a book that's just as advanced. This book explains Linux clearly and effectivelywith a focus on features you care about, from system security and Internet server setup to Windows file/printer sharing. Best-selling author Mark Sobell starts at the beginning and walks you through everything that matters, from installing Linux using the included DVD to working with GNOME, KDE, Samba, **sendmail**, Apache, DNS, NIS, and **iptables**.

This edition contains extensive coverage, including full chapters on using Linux from the command line and GUI; even more thorough system administration and security guidance; and up-to-the-minute, step-by-step instructions for setting up networks and every major type of Internet server. Along the way, you learn the "hows" *and* the "whys." Mark Sobell knows every Linux nook and cranny, has taught hundreds of thousands of readers, and never forgets what it's like to be new to Linux. Whether you are a user, an administrator, or a programmer, this book gives you all you needand more.

Don't settle for yesterday's Linux book...get the ONLY book that meets today's challenges and *tomorrow's!*

Compared with the other Linux books out there, ***A Practical Guide to Red Hat® Linux®, Third Edition,*** delivers...

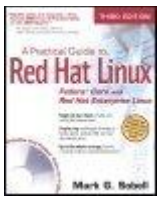
- Complete coverage of Fedora Core and Red Hat Enterprise Linux
- Deeper coverage of the command line and the GNOME and KDE GUIs, including GUI customization
- More practical coverage of file sharing with Samba, NFS, and FTP
- More detailed, usable coverage of Internet server configuration including Apache, **sendmail**, NFS, and DNS/BIND
- More state-of-the-art security techniques, including SELinux (Security Enhanced Linux), ACLs (Access Control Lists), firewall setup using the Red Hat GUI and using **iptables**, and a full chapter on OpenSSH
- More and better coverage of "meat-and-potatoes" system/network administration tasks

- A more practical introduction to writing **bash** shell scripts
- Complete instructions on how to keep your Linux system up-to-date using **yum**
- And much more...including a 500+ term glossary and a comprehensive index to help you find what you need fast!

Includes DVD! Get the full version of Red Hat's Fedora Core 5 release!

NEXT

PREV NEXT



A Practical Guide to Red Hat® Linux®, Third Edition: Fedora™ Core and Red Hat Enterprise Linux

By Mark G. Sobell

.....

Publisher: **Prentice Hall**

Pub Date: **June 27, 2006**

Print ISBN-10: **0-13-228027-2**

Print ISBN-13: **978-0-13-228027-3**

Pages: **1168**

Table of Contents | Index

PREV NEXT

PREV NEXT

Copyright

Library of Congress Cataloging-in-Publication Data:

Sobell, Mark G.
A practical guide to Red Hat Linux : Fedora Core and Red Hat Enterprise Linux / Mark G. Sobell. 3rd ed.
p. cm.
Includes index.
ISBN 0-13-228027-2 (pbk. : alk. paper)
1. Linux. 2. Operating systems (Computers) I. Title.
QA76.76.O63S59485 2006
005'4'32dc22
2006014003

Text printed in the United States on recycled paper at Courier in Stoughton, Massachusetts.

First printing, June 2006

Dedication

For my uncle, David Z. Levitov (1920-2005), who gave me the world.

PREV NEXT

Praise for *A Practical Guide to Red Hat® Linux®, Second Edition*

"Since I'm in an educational environment, I found the content of Sobell's book to be right on target and very helpful for anyone managing Linux in the enterprise. His style of writing is very clear. He builds up to the chapter exercises, which I find to be relevant to real-world scenarios a user or admin would encounter. An IT/IS student would find this book a valuable complement to their education. The vast amount of information is extremely well balanced and Sobell manages to present the content without complicated asides and meandering prose. This is a 'must have' for anyone managing Linux systems in a networked environment or anyone running a Linux server. I would also highly recommend it to an experienced computer user who is moving to the Linux platform."

Mary Norbury
IT Director
Barbara Davis Center/
University of Colorado at Denver
from a review posted on slashdot.org

"I had the chance to use your UNIX books when I when was in college years ago at Cal Poly San Luis Obispo, CA. I have to say that your books are among the best! They're quality books that teach the theoretical aspects and applications of the operating system."

Benton Chan
IS Engineer

"The book has more than lived up to my expectations from the many reviews I read, even though it targets FC2. I have found something very rare with your book: It doesn't read like the standard a technical text, it reads more like a story. It's a pleasure to read and hard to put down. Did I say that?! :-)"

David Hopkins
Business Process Architect

"Thanks for your work and for the book you wrote. There are really few books that can help people to become more efficient administrators of different workstations. We hope (in Russia) that you will continue bringing us a new level of understanding of Linux/UNIX systems."

Anton Petukhov

"Mark Sobell has written a book as approachable as it is authoritative."

Jeffrey Bianchine
Advocate, Author, Journalist

"Excellent reference book, well suited for the sysadmin of a Linux cluster, or the owner of a PC contemplating installing a recent stable Linux. Don't be put off by the daunting heft of the book. Sobell has striven to be as inclusive as possible, in trying to anticipate your system administration needs."

Wes Boudville
Inventor

"*A Practical Guide to Red Hat® Linux®* is a brilliant book. Thank you Mark Sobell."

C. Pozrikidis
University of California at San Diego

"This book presents the best overview of the Linux operating system that I have found. . . . [It] should be very helpful and understandable no matter what the reader's background is: traditional UNIX user, new Linux devotee, or even Windows user. Each topic is presented in a clear, complete fashion and very few assumptions are made about what the reader knows. . . . The book is extremely useful as a reference, as it contains a 70-page glossary of terms and is very well indexed. It is organized in such a way that the reader can focus on simple tasks without having to wade through more advanced topics until they are ready."

Cam Marshall
Marshall Information Service LLC
Member of Front Range UNIX
Users Group [FRUUG]
Boulder, Colorado

"Conclusively, this is THE book to get if you are a new Linux user and you just got into RH/Fedora world. There's no other book that discusses so many different topics and in such depth."

Eugenia Loli-Queru
Editor in Chief
OSNews.com

PREV NEXT

PREV NEXT

Preface

The book

Whether you are an end user, a system administrator, or a little of each, this book explains with step-by-step examples how to get the most out of a Fedora Core or Red Hat Enterprise

Linux system. In 28 chapters, this book takes you from installing a Fedora Core or Red Hat Enterprise Linux system through understanding its inner workings to setting up secure servers that run on the system.

The audience

This book is designed for a wide range of readers. It does not require you to have programming experience, but having some experience using a general-purpose computer is helpful. This book is appropriate for

- **Students** who are taking a class in which they use Linux
- **Home users** who want to set up and/or run Linux
- **Professionals** who use Linux at work
- **System administrators** who need an understanding of Linux and the tools that are available to them
- **Computer science students** who are studying the Linux operating system
- **Programmers** who need to understand the Linux programming environment
- **Technical executives** who want to get a grounding in Linux

Benefits

A Practical Guide to Red Hat® Linux®: Fedora Core™ and Red Hat Enterprise Linux, Third Edition, gives you a broad understanding of many facets of Linux, from installing Red Hat Linux through using and customizing it. No matter what your background, this book gives you the knowledge you need to get on with your work. You will come away from this book understanding how to use Linux, and this book will remain a valuable reference for years to come.

Overlap

If you read *A Practical Guide to Linux® Commands, Editors, and Shell Programming*, you will notice some overlap between that book and the one you are reading now. The first chapter, and the chapters on the utilities, the filesystem, programming tools, and the appendix on regular expressions are very similar in the two books, as are the three chapters on the Bourne Again Shell (bash). Chapters that appear in this book but not in *A Practical Guide to Linux® Commands, Editors, and Shell Programming* include Chapters 2 and 3 (installation), Chapters 4 and 8 (Red Hat Linux and the GUI), Chapter 10 (networking), all of the chapters in Part IV (system administration) and Part V (servers), and Appendix C (security).

This Book Includes Fedora Core 5 on a DVD

A Practical Guide to Red Hat® Linux®, Third Edition, includes a DVD that you can use to install or upgrade to Fedora Core 5. Chapter 2 helps you get ready to install Fedora Core. Chapter 3 provides step-by-step instructions for installing Fedora Core from this DVD. This book guides you through learning about, using, and administrating Fedora Core or Red Hat Enterprise Linux.

What Is New in This Edition?

The third edition of *A Practical Guide to Red Hat® Linux®* covers Fedora Core 5 and Red Hat Enterprise Linux version 4. All the changes, large and small, that have been made to these products since the second edition of this book have been incorporated into the explanations and examples. The following list details the sections of this book that have undergone the most major changes.

- **Access Control Lists** (ACLs; page 185) A security feature that provides finer-grained control over which users can access specific directories and files than do traditional Linux permissions.
- **SELinux** (Security Enhanced Linux; page 400) A security feature that enforces security policies that limit what a user or program can do.
- bash (the Bourne Again Shell; Chapters 7, 9, and 28) These chapters have been reorganized and rewritten to provide clearer explanations and better examples of how bash works both from the command line in day-to-day work and as a programming language to write shell scripts.
- yum (page 476) A program that keeps Fedora Core systems up-to-date. The yum utility downloads software from repositories on the Internet. It can upgrade existing software and install new software. You can run yum manually or have it run automatically every night.
- pirut (page 483) A graphical software package management utility. The pirut utility is similar to yum except that it works with groups of software packages. For example, you can use pirut to download and install the entire KDE desktop environment with one command.
- parted (page 65) A command line utility that reports on and manipulates hard disk partitions.

Features of This Book

This book is designed and organized so you can get the most out of it in the shortest amount of time. You do not have to read this book straight through in page order. Once you are comfortable using Linux, you can use this book as a reference: Look up a topic of interest in the table of contents or index and read about it. Or think of the book as a catalog of Linux topics: Flip through the pages until a topic catches your eye. The book includes many pointers to Web sites where you can get additional information: Consider the Internet an extension of this book.

A Practical Guide to Red Hat® Linux®, Third Edition, is structured with the following features:

- In this book, the term ***Red Hat Linux*** refers to both **Fedora Core** and **Red Hat Enterprise Linux**. Features that apply to only one operating system or the other are marked as such using these indicators: FEDORA or RHEL.
- **Optional sections** enable you to read the book at different levels, returning to more difficult material when you are ready to delve into it.
- **Caution boxes** highlight procedures that can easily go wrong, giving you guidance before you run into trouble.
- **Tip boxes** highlight ways that you can save time by doing something differently or situations when it may be useful or just interesting to have additional information.
- **Security boxes** point out places where you can make a system more secure. The **security appendix** presents a quick background in system security issues.
- Concepts are illustrated by **practical examples** throughout the book.
- **Chapter summaries** review the important points covered in each chapter.
- **Review exercises** are included at the end of each chapter for readers who want to further hone their skills. Answers to even-numbered exercises are at www.sobell.com.

- This book provides resources for **finding software** on the Internet. It also explains how **download** and **install** software using yum, BitTorrent, and, for Red Hat Enterprise Linux, Red Hat Network (RHN).
- The **glossary** defines more than 500 common terms.
- The book describes in detail many important **GNU tools**, including the gcc C compiler, the gdb debugger, the GNU Configure and Build System, make, and gzip.
- Pointers throughout the text provide help in obtaining **online documentation** from many sources including the local system, the Red Hat Web site, and other locations on the Internet.
- Many useful URLs (Internet addresses) point to sites where you can obtain software, security programs and information, and more.
- The comprehensive index helps you locate topics quickly and easily.

Key Topics Covered in This Book

This book contains a lot of information. This section distills and summarizes its contents. You may want to review the table of contents for more detail. This book

Installation

- Describes how to download from the Internet and burn a Fedora Core installation DVD or CDs.
- Helps you plan the layout of the system's hard disk and assists you in using Disk Druid or parted to partition the hard disk.
- Explains how to use the Logical Volume Manager (LVM2) to set up, grow, and migrate logical volumes, which are similar in function to traditional disk partitions.
- Describes in detail how to install Red Hat Linux from a DVD, CDs, a hard disk, or over a network using FTP, NFS, or HTTP.
- Covers responses to the **boot:** prompt and explains how to work with **Anaconda**, Red Hat's installation program.
- Covers the details of installing and customizing the X.org version of the X Window System.

Working with Red Hat Linux

- Introduces the graphical desktop (GUI) and explains how to use desktop tools including the panel, Panel menu, Main menu, Window Operations menu, Desktop menu, Desktop switcher, and terminal emulator.
- Presents the KDE desktop and covers using Konqueror to manage files, start programs, and browse the Web.
- Covers the GNOME desktop and the Nautilus file manager.
- Explains how to customize your desktop to please your senses and help you work more efficiently.
- Covers the Bourne Again Shell (bash) in three chapters, including an entire chapter on shell programming that includes many sample shell scripts.
- Explains the command line interface and introduces more than 30 command line utilities.
- Presents a tutorial on the vim (vi work-alike) textual editor.
- Covers types of networks, network protocols, and network utilities.
- Explains hostnames, IP addresses, and subnets, and explores how to use host and dig to look up domain names and IP addresses on the Internet.

- Covers distributed computing and the client/server model.

System administration

- Explains how to use the Red Hat system-config-* tools to configure the display, DNS, Apache, a network interface, and more. You can also use these tools to add users and manage local and remote printers. (See page 415 for a list of the tools.)
- Describes how to use the following tools to download software and keep a system current:
  - yum Downloads and installs software packages from the Internet, keeping a system up-to-date and resolving dependencies as it processes the packages. You can run yum manually or set it up to run automatically every night.
  - **BitTorrent** Good for distributing large amounts of data such as the Fedora installation DVD and CDs. The more people who use BitTorrent to download a file, the faster it works.
  - up2date The Red Hat Enterprise Linux tool for keeping system software current.
- Covers graphical system administration tools, including the Main menu, GNOME and KDE menu systems, KDE Control Center, and KDE Control panel.
- Explains system operation, including the boot process, init scripts, emergency mode, rescue mode, single-user and multiuser modes, and steps to take if the system crashes.
- Describes files, directories, and filesystems, including types of files and filesystems, **fstab** (the filesystem table), automatically mounted filesystems, filesystem integrity checks, filesystem utilities, and fine-tuning of filesystems.
- Covers backup utilities including tar, cpio, dump, and restore.
- Explains how to customize and build a Linux kernel.

Security

- Helps you manage basic system security issues using ssh (secure shell), **vsftpd** (secure FTP server), Apache (the **httpd** Web server), iptables (firewall), and more.
- Presents a complete section on SELinux (Security Enhanced Linux), including instructions for using system-config-securitylevel to configure SELinux.
- Covers using system-config-securitylevel to set up a basic firewall to protect the system.
- Provides instructions on using iptables to share an Internet connection over a LAN and to build advanced firewalls.
- Describes how to set up a chroot jail to protect a server system.
- Explains how to use TCP wrappers to control who can access a server.
- Covers controlling servers using the **xinetd** superserver.

Clients and servers

- Explains how to set up and use the most popular Linux servers, providing a chapter on each: Apache, Samba, OpenSSH, **sendmail**, DNS, NFS, FTP, iptables, and NIS (all of which are included with Red Hat Linux).
- Describes how to set up a CUPS printer server.
- Describes how to set up and use a DHCP server.

- Covers programming tools including the GNU gcc compiler, the gdb debugger, make, and CVS for managing source code.
- Explains how to debug a C program.
- Describes how to work with shared libraries.
- Provides a complete chapter on shell programming using bash, including many examples.

Details

Part I

Part I, "Installing Red Hat Linux," discusses how to install Fedora Core or Red Hat Enterprise Linux. **Chapter 2** presents an overview of the process of installing Red Hat Linux, including hardware requirements, downloading and burning a DVD or CDs, and planning the layout of the hard disk. **Chapter 3** is a step-by-step guide to installing either version of Red Hat Linux and covers installing from a DVD or CDs, from a local hard disk, and over the network using FTP, NFS, or HTTP. It also shows how to set up the X Window System and customize your graphical user interface (GUI).

Part II

Part II, "Getting Started with Red Hat Linux," familiarizes you with Red Hat Linux, covering logging in, the GUI, utilities, the filesystem, and the shell. **Chapter 4** introduces desktop features, including the panel and the Main menu; explains how to use Konqueror to manage files, run programs, and browse the Web; and covers finding documentation, dealing with login problems, and using the window manager. **Chapter 5** introduces the shell command line interface, describes more than 30 useful utilities, and presents a tutorial on the vim text editor. **Chapter 6** discusses the Linux hierarchical filesystem, covering files, filenames, pathnames, working with directories, access permissions, and hard and symbolic links. **Chapter 7** introduces the Bourne Again Shell (bash) and discusses command line arguments and options, redirecting input to and output from commands, running programs in the background, and using the shell to generate and expand filenames.

Tip: Experienced users may want to skim Part II

If you have used a UNIX or Linux system before, you may want to skim over or skip some or all of the chapters in Part II. All readers should take a look at "Conventions Used in This Book" (page 17), which explains the typographic and layout conventions that this book uses, and "Getting the Facts: Where to Find Documentation" (page 102), which points out both local and remote sources of Linux and Red Hat documentation.

Part III

Part III, "Digging into Red Hat Linux," goes into more detail about working with the system. **Chapter 8** discusses the GUI and includes a section on how to run a graphical program on a remote system and have the display appear locally. The section on GNOME describes GNOME utilities and explains how to use the Nautilus file manager, including its spatial view, while the section on KDE explains more about Konqueror and KDE utilities. **Chapter 9** extends the bash coverage from Chapter 7, explaining how to redirect error output, avoid overwriting files, and work with job control, processes, startup files, important shell builtin commands, parameters, shell variables, and aliases. **Chapter 10** explains networks, network security, and the Internet and discusses types of networks, subnets, protocols, addresses, hostnames, and various network utilities. The section on distributed computing describes the client/server model and some of the servers you can use on a network. Details of setting up and using clients and servers are reserved until Part V.

Part IV

Part IV covers system administration. **Chapter 11** discusses core concepts such as Superuser, SELinux (Security Enhanced Linux), system operation, general information about how to set up a server, DHCP, and PAM. **Chapter 12** explains the Linux filesystem, going into detail about types of files, including special and device files, the use of fsck to verify the integrity of and repair filesystems, and the use of tune2fs to change filesystem parameters. **Chapter 13** explains how to keep a system up-to-date by downloading software from the Internet and installing it, including examples of using yum, BitTorrent, and Red Hat's up2date utility. **Chapter 14** explains how to set up the CUPS printing system so you can print on the local system as well as on remote systems. **Chapter 15** details customizing and building a Linux kernel. **Chapter 16** covers additional administration tasks, including setting up user accounts, backing up files, scheduling automated tasks, tracking disk usage, and solving general problems. **Chapter 17** explains how to set up a local area network (LAN), including both hardware (including wireless) and software setup.

Part V

Part V goes into detail about setting up and running servers and connecting to them with clients. The chapters in this part of the book cover the following clients/servers:

- **OpenSSH** Set up an OpenSSH server and use sh, scp, and sftp to communicate securely over the Internet.
- **FTP** Set up a **vsftpd** secure FTP server and use any of several FTP clients to exchange files with the server.
- **Mail** Configure **sendmail** and use Webmail, POP3, or IMAP to retrieve email; use SpamAssassin to combat spam.
- **NIS** Set up NIS to facilitate system administration of a LAN.
- **NFS** Share filesystems between systems on a network.
- **Samba** Share filesystems and printers between Windows and Linux systems.
- **DNS/BIND** Set up a domain nameserver to let other systems on the Internet know the names and IP addresses of your systems they may need to contact.
- **iptables** Share a single Internet connection between systems on a LAN and set up a

firewall to protect local systems.
- **Apache** Set up an HTTP server that serves Web pages that browsers can display.

Part VI

Part VI covers programming. **Chapter 27** discusses programming tools and environments available under Red Hat Linux, including the C programming language and debugger, make, shared libraries, and source code management using CVS. **Chapter 28** goes into greater depth about shell programming using bash, with the discussion being enhanced by extensive examples.

Part VII

Part VII includes appendixes on regular expressions, helpful Web sites, system security, and free software. This part also includes an extensive glossary with more than 500 entries and a comprehensive index.

Supplements

The author's home page (www.sobell.com) contains downloadable listings of the longer programs from this book as well as pointers to many interesting and useful Linux sites on the World Wide Web, a list of corrections to the book, answers to even-numbered exercises, and a solicitation for corrections, comments, and suggestions.

Thanks

First and foremost I want to thank Mark L. Taub, editor-in-chief, Prentice Hall, who encouraged and prodded me (carrot-and-stick approach) and kept me on track. Mark is unique in my 25 years of book writing experience: an editor who works with the tools I am writing about. Because Mark runs Linux on his home computer, we share experiences as I write. His comments and direction are invaluable. Thank you, Mark T.

Thanks also to the folks at Prentice Hall who helped bring this book to life, especially Julie Nahil, full-service production manager, who gave me guidance and much latitude while keeping me to schedule in producing the book; John Fuller, managing editor, who kept the large view in check; Noreen Regina, editorial assistant, who attended to the many details involved in publishing this book; Heather Fox, publicist; Dan Scherf, media developer; Sandra Schroeder, design manager; Kim Spilker, marketing manager; and everyone else who worked behind the scenes to make this book happen.

I am also indebted to Denis Howe, the editor of *The Free On-line Dictionary of Computing* (FOLDOC). Denis has graciously permitted me to use entries from his compilation. Be sure to look at the dictionary (www.foldoc.org).

A big "thank you" to the folks who read through the drafts of the book and made comments that caused me to refocus parts of the book where things were not clear or were left out altogether: David Chisnall; Chris Karr, Northwestern University; Jesse Keating, Fedora Project; Scott Mann, IBM, Systems Management and Integration Professional; Matthew Miller, Boston University; and George Vish, Senior Education Consultant, HP US Linux

Program Manager, Hewlett-Packard Company.

Thanks also to the following people who helped with the first and second editions of this book: Carsten Pfeiffer, Software Engineer and KDE Developer; Aaron Weber, Ximian; Cristof Falk, Software Developer at CritterDesign; Steve Elgersma, Computer Science Department, Princeton University; Scott Dier, University of Minnesota; Robert Haskins, Computer Net Works; Lars Kellogg-Stedman, Harvard University; Jim A. Lola, Principal Systems Consultant, Privateer Systems, LLC; Eric S. Raymond, cofounder, Open Source Initiative; Scott Mann; Randall Lechlitner, Independent Computer Consultant; Jason Wertz, Computer Science Instructor, Montgomery County Community College; Justin Howell, Solano Community College; Ed Sawicki, The Accelerated Learning Center; David Mercer, Contechst; Jeffrey Bianchine, Advocate, Author, Journalist; John Kennedy; and Jim Dennis, Starshine Technical Services.

Thanks also to Dustin Puryear, Puryear Information Technology; Gabor Liptak, Independent Consultant; Bart Schaefer, Chief Technical Officer, iPost; Michael J. Jordan, Web Developer, Linux Online Inc.; Steven Gibson, owner of SuperAnt.com; John Viega, founder and Chief Scientist, Secure Software, Inc.; K. Rachael Treu, Internet Security Analyst, Global Crossing; Kara Pritchard, K & S Pritchard Enterprises, Inc.; Glen Wiley, Capital One Finances; Karel Baloun, Senior Software Engineer, Looksmart, Ltd.; Matthew Whitworth; Dameon D. Welch-Abernathy, Nokia Systems; Josh Simon, Consultant; Stan Isaacs; and Dr. Eric H. Herrin II, Vice President, Herrin Software Development, Inc. And thanks to Doug Hughes, long-time system designer and administrator, who gave me a big hand with the sections on system administration, networks, the Internet, and programming.

More thanks go to consultants Lorraine Callahan and Steve Wampler; Ronald Hiller, Graburn Technology, Inc.; Charles A. Plater, Wayne State University; Bob Palowoda; Tom Bialaski, Sun Microsystems; Roger Hartmuller, TIS Labs at Network Associates; Kaowen Liu; Andy Spitzer; Rik Schneider; Jesse St. Laurent; Steve Bellenot; Ray W. Hiltbrand; Jennifer Witham; Gert-Jan Hagenaars; and Casper Dik.

A Practical Guide to Red Hat® Linux®, Third Edition, is based in part on two of my previous UNIX books: *UNIX System V: A Practical Guide* and *A Practical Guide to the UNIX System*. Many people helped me with those books, and thanks here go to Pat Parseghian, Dr. Kathleen Hemenway, and Brian LaRose; Byron A. Jeff, Clark Atlanta University; Charles Stross; Jeff Gitlin, Lucent Technologies; Kurt Hockenbury; Maury Bach, Intel Israel Ltd.; Peter H. Salus; Rahul Dave, University of Pennsylvania; Sean Walton, Intelligent Algorithmic Solutions; Tim Segall, Computer Sciences Corporation; Behrouz Forouzan, DeAnza College; Mike Keenan, Virginia Polytechnic Institute and State University; Mike Johnson, Oregon State University; Jandelyn Plane, University of Maryland; Arnold Robbins and Sathis Menon, Georgia Institute of Technology; Cliff Shaffer, Virginia Polytechnic Institute and State University; and Steven Stepanek, California State University, Northridge, for reviewing the book.

I continue to be grateful to the many people who helped with the early editions of my UNIX books. Special thanks are due to Roger Sippl, Laura King, and Roy Harrington for introducing me to the UNIX system. My mother, Dr. Helen Sobell, provided invaluable comments on the original manuscript at several junctures. Also, thanks go to Isaac Rabinovitch, Professor Raphael Finkel, Professor Randolph Bentson, Bob Greenberg, Professor Udo Pooch, Judy Ross, Dr. Robert Veroff, Dr. Mike Denny, Joe DiMartino, Dr. John Mashey, Diane Schulz, Robert Jung, Charles Whitaker, Don Cragun, Brian Dougherty, Dr. Robert Fish, Guy Harris, Ping Liao, Gary Lindgren, Dr. Jarrett Rosenberg, Dr. Peter Smith, Bill Weber, Mike Bianchi, Scooter Morris, Clarke Echols, Oliver Grillmeyer, Dr. David Korn, Dr. Scott Weikart, and Dr. Richard Curtis.

I take responsibility for any errors and omissions in this book. If you find one or just have a comment, let me know (mgs@sobell.com) and I will fix it in the next printing. My home page (www.sobell.com) contains a list of errors and credits those who found them. It also

offers copies of the longer scripts from the book and pointers to many interesting Linux pages.

Mark G. Sobell
San Francisco, California

PREV NEXT

PREV NEXT

1. Welcome to Linux

IN THIS CHAPTER

The Linux *kernel* was developed by Finnish undergraduate student Linus Torvalds, who used the Internet to make the source code immediately available to others for free. Torvalds released Linux version 0.01 in September 1991.

The new operating system came together through a lot of hard work. Programmers around the world were quick to extend the kernel and develop other tools, adding functionality to match that already found in both BSD UNIX and System V UNIX (SVR4) as well as new functionality.

The Linux operating system, developed through the cooperation of many, many people around the world, is a *product of the Internet* and is a *free* operating system. In other words, all the source code is free. You are free to study it, redistribute it, and modify it. As a result, the code is available free of costno charge for the software, source, documentation, or support (via newsgroups, mailing lists, and other Internet resources). As the GNU Free Software Definition (reproduced in Appendix D) puts it:

> Free beer
>
> "Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer."

The GNULinux Connection

An operating system is the low-level software that schedules tasks, allocates storage, and handles the interfaces to peripheral hardware, such as printers, disk drives, the screen, keyboard, and mouse. An operating system has two main parts: the *kernel* and the *system programs.* The kernel allocates machine resources, including memory, disk space, and *CPU* (page 1026) cycles, to all other programs that run on the computer. The system programs perform higher-level housekeeping tasks, often acting as servers in a client/server relationship. *Linux* is the name of the kernel that Linus Torvalds presented to the world in 1991 and that many others have worked on since then to enhance, stabilize, expand, and make more secure.

The History of GNULinux

This section presents some background on the relationship between GNU and Linux.

Fade to 1983

Richard Stallman (www.stallman.org) announced[1] the GNU Project for creating an operating system, both kernel and system programs, and presented the GNU Manifesto,[2] which begins as follows:

[1] www.gnu.org/gnu/initial-announcement.html

[2] www.gnu.org/gnu/manifesto.html

> GNU, which stands for Gnu's Not UNIX, is the name for the complete UNIX-compatible software system which I am writing so that I can give it away free to everyone who can use it.

Some years later, Stallman added a footnote to the preceding sentence when he realized that it was creating confusion:

> The wording here was careless. The intention was that nobody would have to pay for *permission* to use the GNU system. But the words don't make this clear, and people often interpret them as saying that copies of GNU should always be distributed at little or no charge. That was never the intent; later on, the manifesto mentions the possibility of companies providing the service of distribution for a profit. Subsequently I have learned to distinguish carefully between "free" in the sense of freedom and "free" in the sense of price. Free software is software that users have the freedom to distribute and change. Some users may obtain copies at no charge, while others pay to obtain copiesand if the funds help support improving the software, so much the better. The important thing is that everyone who has a copy has the freedom to cooperate with others in using it.

In the manifesto, after explaining a little about the project and what has been accomplished

so far, Stallman continues:

> **Why I Must Write GNU**
>
> I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. For years I worked within the Artificial Intelligence Lab to resist such tendencies and other inhospitalities, but eventually they had gone too far: I could not remain in an institution where such things are done for me against my will.
>
> So that I can continue to use computers without dishonor, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free. I have resigned from the AI Lab to deny MIT any legal excuse to prevent me from giving GNU away.

Next Scene, 1991

The GNU Project has moved well along toward its goal. Much of the GNU operating system, except for the kernel, is complete. Richard Stallman later writes:

> By the early '90s we had put together the whole system aside from the kernel (and we were also working on a kernel, the GNU Hurd,[3] which runs on top of Mach[4]). Developing this kernel has been a lot harder than we expected, and we are still working on finishing it.[5]
>
> ...[M]any believe that once Linus Torvalds finished writing the kernel, his friends looked around for other free software, and for no particular reason most everything necessary to make a UNIX-like system was already available.
>
> What they found was no accidentit was the GNU system. The available free software[6] added up to a complete system because the GNU Project had been working since 1984 to make one. The GNU Manifesto had set forth the goal of developing a free UNIX-like system, called GNU. The Initial Announcement of the GNU Project also outlines some of the original plans for the GNU system. By the time Linux was written, the [GNU] system was almost finished.[7]
>
> [3] www.gnu.org/software/hurd/hurd.html
>
> [4] www.gnu.org/software/hurd/gnumach.html
>
> [5] www.gnu.org/software/hurd/hurd-and-linux.html
>
> [6] See Appendix D or www.gnu.org/philosophy/free-sw.html.
>
> [7] www.gnu.org/gnu/linux-and-gnu.html

Today the GNU "operating system" runs on top of the FreeBSD (www.freebsd.org) and NetBSD (www.netbsd.org) kernels with complete Linux binary compatibility and on top of Hurd pre-releases and Darwin (developer.apple.com/opensource) without this compatibility.

The Code Is Free

The tradition of free software dates back to the days when UNIX was released to universities at nominal cost, which contributed to its portability and success. This tradition died as UNIX was commercialized and manufacturers regarded the source code as

proprietary, making it effectively unavailable. Another problem with the commercial versions of UNIX related to their complexity. As each manufacturer tuned UNIX for a specific architecture, it became less portable and too unwieldy for teaching and experimentation.

MINIX

Two professors created their own stripped-down UNIX look-alikes for educational purposes: Doug Comer created XINU (www.cs.purdue.edu/research/xinu.html) and Andrew Tanenbaum created MINIX (www.cs.vu.nl/~ast/minix.html). Linus Torvalds created Linux to counteract the shortcomings in MINIX. Every time there was a choice between code simplicity and efficiency/features, Tanenbaum chose simplicity (to make it easy to teach with MINIX), which meant that this system lacked many features people wanted. Linux goes in the opposite direction.

You can obtain Linux at no cost over the Internet (page 35). You can also obtain the GNU code via the U.S. mail at a modest cost for materials and shipping. You can support the Free Software Foundation (www.fsf.org) by buying the same (GNU) code in higher-priced packages, and you can buy commercial packaged releases of Linux (called *distributions*), such as Red Hat Linux, that include installation instructions, software, and support.

GPL

Linux and GNU software are distributed under the terms of the GNU General Public License (GPL, www.gnu.org/licenses/licenses.html). The GPL says you have the right to copy, modify, and redistribute the code covered by the agreement. When you redistribute the code, however, you must also distribute the same license with the code, making the code and the license inseparable. If you get source code off the Internet for an accounting program that is under the GPL and then modify that code and redistribute an executable version of the program, you must also distribute the modified source code and the GPL agreement with it. Because this arrangement is the reverse of the way a normal copyright works (it gives rights instead of limiting them), it has been termed a *copyleft*. (This paragraph is not a legal interpretation of the GPL; it is here merely to give you an idea of how it works. Refer to the GPL itself when you want to make use of it.)

Have Fun!

Two key words for Linux are "Have Fun!" These words pop up in prompts and documentation. The UNIXnow Linuxculture is steeped in humor that can be seen throughout the system. For example, `less` is `more`GNU has replaced the UNIX paging utility named `more` with an improved utility named `less`. The utility to view PostScript documents is named `ghostscript`, and one of several replacements for the `vi` editor is named `elvis`. While machines with Intel processors have "Intel Inside" logos on their outside, some Linux machines sport "Linux Inside" logos. And Torvalds himself has been seen wearing a T-shirt bearing a "Linus Inside" logo.

The Linux 2.6 Kernel

The Linux 2.6 kernel was released on December 17, 2003. This kernel has many features that offer increased security and speed. Some of these features benefit end users directly; others help developers produce better code and find problems more quickly. See Appendix E for a description of the new features in the Linux 2.6 kernel.

PREV NEXT

The Heritage of Linux: UNIX

The UNIX system was developed by researchers who needed a set of modern computing tools to help them with their projects. The system allowed a group of people working together on a project to share selected data and programs while keeping other information private.

Universities and colleges played a major role in furthering the popularity of the UNIX operating system through the "four-year effect." When the UNIX operating system became widely available in 1975, Bell Labs offered it to educational institutions at nominal cost. The schools, in turn, used it in their computer science programs, ensuring that computer science students became familiar with it. Because UNIX was such an advanced development system, the students became acclimated to a sophisticated programming environment. As these students graduated and went into industry, they expected to work in a similarly advanced environment. As more of them worked their way up the ladder in the commercial world, the UNIX operating system found its way into industry.

In addition to introducing students to the UNIX operating system, the Computer Systems Research Group (CSRG) at the University of California at Berkeley made significant additions and changes to it. In fact, it made so many popular changes that one version of the system is called the Berkeley Software Distribution (BSD) of the UNIX system (or just Berkeley UNIX). The other major version is UNIX System V (SVR4), which descended from versions developed and maintained by AT&T and UNIX System Laboratories.

What Is So Good About Linux?

In recent years Linux has emerged as a powerful and innovative UNIX work-alike. Its popularity is surpassing that of its UNIX predecessors. Although it mimics UNIX in many ways, the Linux operating system departs from UNIX in several significant ways: The Linux kernel is implemented independently of both BSD and System V, the continuing development of Linux is taking place through the combined efforts of many capable individuals throughout the world, and Linux puts the power of UNIX within easy reach of business and personal computer users. Using the Internet, today's skilled programmers submit additions and improvements to the operating system to Linus Torvalds, GNU, or one of the other authors of Linux.

Applications

A rich selection of applications is available for Linuxboth free and commercialas well as a wide variety of tools: graphical, word processing, networking, security, administration, Web server, and many others. Large software companies have recently seen the benefit in supporting Linux and now have on-staff programmers whose job it is to design and code the Linux kernel, GNU, KDE, or other software that runs on Linux. For example, IBM (www.ibm.com/linux) is a major Linux supporter. Linux conforms increasingly more closely to POSIX standards, and some distributions and parts of others meet this standard. (See "Standards" on page 8 for more information.) These developments mean that Linux is becoming more mainstream and is respected as an attractive alternative to other popular operating systems.

Peripherals

Another aspect of Linux that appeals to users is the amazing range of peripherals that is supported and the speed with which support for new peripherals emerges. Linux often supports a peripheral or interface card before any company does. Unfortunately some types of peripheralsparticularly proprietary graphics cardslag in their support because the manufacturers do not release specifications or source code for drivers in a timely manner, if at all.

Software

Also important to users is the amount of software that is availablenot just source code (which needs to be compiled) but also prebuilt binaries that are easy to install and ready to run. These include more than free software. Netscape, for example, has been available for Linux from the start and included Java support before it was available from many commercial vendors. Now its sibling Mozilla/Thunder-bird/Firefox is also a viable browser, mail client, and newsreader, performing many other functions as well.

Platforms

Linux is not just for Intel-based platforms: It has been ported to and runs on the Power PCincluding Apple computers (ppclinux), Compaq's (née Digital Equipment Corporation) Alpha-based machines, MIPS-based machines, Motorola's 68K-based machines, various 64-bit systems, and IBM's S/390. Nor is Linux just for single-processor machines: As of version 2.0, it runs on multiple-processor machines (SMPs). It also includes an O(1) scheduler, which dramatically increases scalability on SMP systems.

Emulators

Linux supports programs, called *emulators*, that run code intended for other operating systems. By using emulators you can run some DOS, Windows, and Macintosh programs under Linux. Wine (www.winehq.com) is an open-source implementation of the Windows API on top of the X Window System and UNIX/Linux; QEMU (fabrice.bellard.free.fr/qemu) is a CPU-only emulator that executes x86 Linux binaries on non-x86 Linux systems.

Xen

Xen, which was created at the University of Cambridge and is now being developed in the open-source community, is an open-source virtual machine monitor (VMM). A VMM enables several virtual machines (VMs), each running an instance of a separate operating system, to run on a single computer. Xen isolates the VMs so that if one crashes it does not affect any of the others. In addition, Xen introduces minimal performance overhead when compared with running each of the operating systems natively.

Using VMs, you can experiment with cutting-edge releases of operating systems and applications without concern for the base (stable) system, all on a single machine. You can also set up and test networks of systems on a single machine. Xen presents a *sandbox*, an area (system) that you can work in without regard for the results of your work or for the need to clean up.

Fedora Core 5 includes Xen 3.0. This book does not cover the installation or use of Xen. See www.fedoraproject.org/wiki/FedoraXenQuickstartFC5 for installation instructions.

For more information on Xen, refer to the wiki at wiki.xensource.com/xenwiki and the Xen home page at www.cl.cam.ac.uk/Research/SRG/netos/xen.

Why Linux Is Popular with Hardware Companies and Developers

Two trends in the computer industry set the stage for the popularity of UNIX and Linux. First, advances in hardware technology created the need for an operating system that could take advantage of available hardware power. In the mid-1970s, minicomputers began challenging the large mainframe computers because, in many applications, minicomputers could perform the same functions less expensively. More recently, powerful 64-bit processor chips, plentiful and inexpensive memory, and lower-priced hard disk storage have allowed hardware companies to install multiuser operating systems on desktop computers.

Proprietary operating systems

Second, with the cost of hardware continually dropping, hardware manufacturers could no longer afford to develop and support proprietary operating systems. A *proprietary* operating system is written and owned by the manufacturer of the hardware (for example, DEC/Compaq owns VMS). Today's manufacturers need a generic operating system that they can easily adapt to their machines.

Generic operating systems

A *generic* operating system is written outside of the company manufacturing the hardware and is sold (UNIX, Windows) or given (Linux) to the manufacturer. Linux is a generic operating system because it runs on different types of hardware produced by different manufacturers. Of course, if manufacturers can pay only for development and avoid per-unit costs (as they have to pay to Microsoft for each copy of Windows they sell), manufacturers are much better off. In turn, software developers need to keep the prices of their products down; they cannot afford to convert their products to run under many different proprietary operating systems. Like hardware manufacturers, software developers need a generic operating system.

Although the UNIX system once met the needs of hardware companies and researchers for a generic operating system, over time it has become more proprietary as each manufacturer added support for specialized features and introduced new software libraries and utilities.

Linux emerged to serve both needs. It is a generic operating system that takes advantage of available hardware power.

Linux Is Portable

A *portable* operating system is one that can run on many different machines. More than 95 percent of the Linux operating system is written in the C programming language, and C is portable because it is written in a higher-level, machine-independent language. (The C compiler is written in C.)

Because Linux is portable, it can be adapted (ported) to different machines and can meet special requirements. For example, Linux is used in embedded computers, such as the ones found in cellphones, PDAs, and the cable boxes on top of many TVs. The file structure takes full advantage of large, fast hard disks. Equally important, Linux was originally designed as a multiuser operating systemit was not modified to serve several users as an afterthought. Sharing the computer's power among many users and giving them the ability to share data and programs are central features of the system.

Because it is adaptable and takes advantage of available hardware, Linux runs on many different microprocessor-based systems as well as mainframes. The popularity of the microprocessor-based hardware drives Linux; these microcomputers are getting faster all the time, at about the same price point. Linux on a fast microcomputer has become good enough to displace workstations on many desktops. Linux benefits both users, who do not like having to learn a new operating system for each vendor's hardware, and system administrators, who like having a consistent software environment.

The advent of a standard operating system has aided the development of the software industry. Now software manufacturers can afford to make one version of a product available on machines from different manufacturers.

Standards

Individuals from companies throughout the computer industry have joined together to develop the POSIX (Portable Operating System Interface for Computer Environments) standard, which is based largely on the UNIX System V Interface Definition (SVID) and other earlier standardization efforts. These efforts have been spurred by the U.S. government, which needs a standard computing environment to minimize its training and procurement costs. Now that these standards are gaining acceptance, software developers are able to develop applications that run on all conforming versions of UNIX, Linux, and other operating systems.

The C Programming Language

Ken Thompson wrote the UNIX operating system in 1969 in PDP-7 assembly language. Assembly language is machine dependent: Programs written in assembly language work on only one machine or, at best, one family of machines. The original UNIX operating system therefore could not easily be transported to run on other machines (it was not portable).

To make UNIX portable, Thompson developed the B programming language, a machine-independent language, from the BCPL language. Dennis Ritchie developed the C programming language by modifying B and, with Thompson, rewrote UNIX in C in 1973. The revised operating system could be transported more easily to run on other machines.

That development marked the start of C. Its roots reveal some of the reasons why it is such a powerful tool. C can be used to write machine-independent programs. A programmer who designs a program to be portable can easily move it to any computer that has a C compiler. C is also designed to compile into very efficient code. With the advent of C, a programmer no longer had to resort to assembly language to get code that would run well (that is, quicklyalthough an assembler will always generate more efficient code than a high-level language).

C is a good systems language. You can write a compiler or an operating system in C. It is highly structured but is not necessarily a high-level language. C allows a programmer to manipulate bits and bytes, as is necessary when writing an operating system. But it also has high-level constructs that allow efficient, modular programming.

In the late 1980s the American National Standards Institute (ANSI) defined a standard version of the C language, commonly referred to as *ANSI C* or *C89* (for the year the standard was published). Ten years later the C99 standard was published; it is mostly supported by the GNU Project's C compiler (named `gcc`). The original version of the language is often referred to as *Kernighan & Ritchie* (or *K&R*) C, named for the authors of the book that first described the C language.

Another researcher at Bell Labs, Bjarne Stroustrup, created an object-oriented programming language named *C++*, which is built on the foundation of C. Because object-oriented programming is desired by many employers today, C++ is preferred over C in many environments. Another language of choice is Objective-C, which was used to write the first Web browser. The GNU Project's C compiler supports C, C++, and Objective-C.

Skipped, please register!

Ex Libris Eugeen A. Katkovsky